


Article

Evolving Hierarchical and Tag Information via the Deeply Enhanced Weighted Non-Negative Matrix Factorization of Rating Predictions

Alpamis Kutlimuratov ¹, Akmalbek Abdusalomov ¹  and Taeg Keun Whangbo ^{2,*}

¹ Department of IT Convergence Engineering, Gachon University, Sujeong-Gu, Seongnam-Si, Gyeonggi-Do 461-701, Korea; alpamis92@gc.gachon.ac.kr (A.K.); akmaljon@gachon.ac.kr (A.A.)

² Department of Computer Science, Gachon University, Sujeong-Gu, Seongnam-Si, Gyeonggi-Do 461-701, Korea

* Correspondence: tkwhangbo@gachon.ac.kr

Received: 5 October 2020; Accepted: 18 November 2020; Published: 23 November 2020



Abstract: Identifying the hidden features of items and users of a modern recommendation system, wherein features are represented as hierarchical structures, allows us to understand the association between the two entities. Moreover, when tag information that is added to items by users themselves is coupled with hierarchically structured features, the rating prediction efficiency and system personalization are improved. To this effect, we developed a novel model that acquires hidden-level hierarchical features of users and items and combines them with the tag information of items that regularizes the matrix factorization process of a basic weighted non-negative matrix factorization (WNMF) model to complete our prediction model. The idea behind the proposed approach was to deeply factorize a basic WNMF model to obtain hidden hierarchical features of user's preferences and item characteristics that reveal a deep relationship between them by regularizing the process with tag information as an auxiliary parameter. Experiments were conducted on the MovieLens 100K dataset, and the empirical results confirmed the potential of the proposed approach and its superiority over models that use the primary features of users and items or tag information separately in the prediction process.

Keywords: recommendation system; weighted non-negative matrix factorization; hierarchical information; tag information; deep factorization

1. Introduction

Recently, with the increase in the availability of data from online content providers, delivering valuable information that gratifies and holds a consumer's interest has attracted significant attention; thus, modeling an effective recommendation system is essential. The primary objective of a recommendation system is to offer suggestions based on user preferences, which are solicited from historical data, such as ratings, reviews, and tags. Recommendations help in accelerating searches and enable users to access more pertinent content. Therefore, web service providers have extensively cogitated about developing recommender systems that analyze and harness user-item interactions to increase customer satisfaction, profits, and personalized suggestions for their services. Several modern-day internet applications have integrated recommendation systems, including Google, Netflix, eBay, and Amazon.

Recommendation systems are designed based on the type of information obtained such that the diversity of information influences their implementation and structure. To this effect, two traditional approaches exist for building recommendation systems: content-based filtering (CBF) and collaborative

filtering (CF) [1,2]. The former approach generates recommendations by analyzing the availability of the user–item interaction data, which largely requires collecting explicit information [3–5]. For instance, content-based movie recommendations accommodate the features of a movie that match those of a user’s past preferences. Thus, identifying a connection between the items and users is highly important. However, in recent years, owing to the limitations of this approach, such as privacy concerns and the dearth of supplementary information for items, web services have adopted the CF architecture in recommendation systems. The algorithms of this method utilize the items rated by a user to predict unrated items when offering recommendations and subsequently automate these predictions by acquiring user perceptions among a niche audience [2,3,6–10].

Memory- and model-based techniques are commonly used to elucidate CF recommendations [3,11–15]. Past studies have demonstrated the benefits of memory-based CF, wherein rating predictions are computed from the preferences of similar users via a rating matrix [12,16–19]. Conversely, the model-based CF technique leverages a user–item rating matrix to initially build a predictive model using deep learning methods and then source the rating predictions from it [3,20]. CF-based recommendation systems are susceptible to data sparsity and the cold-start problem, which are open issues in the recommendation system research area and put the responsibility on any kind of recommendation system algorithms and methods to avoid and solve them [21]. First, fewer user interactions with items in a user–item rating matrix invokes data sparsity; specifically, the input rating matrix is not sufficient to train a model to make predictions. Thus, only 10–25% of the matrix is populated with ratings. Second, the cold-start problem arises when information about new users or items and their interactions is insufficient to garner suitable recommendations.

One of the most effective implementations of model-based CF is the matrix factorization (MF) method. This method deconstructs the user–item rating matrix into two less latent factor sub-matrices of user preferences and item characteristics, respectively, and then a vector constituting an item and a user feature is generated to predict the user’s rating for an item [3,14,22,23]. Moreover, MF spontaneously integrates a mix of implicit and explicit information related to users or items. Factorization methods have since demonstrated substantial efficiency when resolving the issues of data sparsity and the cold start in recommendation systems.

In this study, we aimed to address the two aforementioned issues using hierarchical and tag information through enhanced matrix factorization to eventually improving the performance of recommender systems. Hierarchical information helps with meaningfully concealing information regarding items, such as categories of movie genres on streaming websites (e.g., Netflix and Disney+) or product catalogs on shopping websites (e.g., Amazon, Alibaba, and eBay).

Users and items of the real practical recommendation systems could exhibit certain hierarchical structures. For example, a user (girl) may usually select movies from the main category “romance,” or more exactly, the user watches movies under the sub-category of romance drama. Similarly, the item (the Apple Watch Series 5) can be placed in the main category “electronics,” or more specifically, the item is tantamount to the sub-category “smart watches.” The classification of an item into appropriate lower-level categories or nodes is conducted sequentially. Items in the same hierarchical level are likely to share similar attributes, thus they are likely to get similar rating scores. Equivalently, users in the same hierarchical level are likely to share similar preferences, thus they are likely to rate certain items similarly [24]. For this reason, recently, evolving hierarchical structures of items or users have been developing to improve recommendation system performances. The priority of hierarchical structures and their unavailability also motivated us to research hierarchical structures of users and items for recommendation systems. During the research, evolving the hierarchical structures of items and users simultaneously and mathematically modeling them for recommendation systems were studied. Along with the above, integrating tag information with mathematically modeled hierarchical structures of items and users into a systematic model that puts a basis for a recommendation system was also investigated.

In contrast, tag information comprises words or short phrases assigned to items by a user that reflects their associations or behavior, and in turn, facilitates predictions by passing it as a value to the prediction algorithm. Researchers have previously reported on the benefits of making recommendations using tags and generating hierarchical information to not only improve results but also tackle issues of data sparsity and cold starts [8,13,24–28]. Furthermore, to the best of our knowledge, despite the significant amount of research that has been conducted to explicate the use of matrix factorization via hierarchical and tag information individually in recommendation systems, the two have rarely been applied in a combination.

In this study, we developed a novel MF-based methodology to predict ratings by incorporating both hierarchical and tag information simultaneously. The rationale behind the proposed approach was to deeply enrich a basic MF model to obtain hierarchical relationships for predicting the ratings and then regularize it using tag information. Our main contributions using this approach included the following:

- Deeply extending the basic MF model to identify hierarchical relationships that facilitate the rating predictions.
- Regularizing the resultant model with tag information, as well as hierarchical data.
- Conducting experiments on the MovieLens 100K dataset (<https://grouplens.org/datasets/movielens/>) to evaluate the proposed methodology.
- Reducing data sparsity and cold-start issues encountered by other CF methods.

The remainder of this paper is structured as follows. In Section 2, works pertaining to tag-based recommendation systems, generating hierarchical features, and existing MF methods are reviewed. In Sections 3 and 4, we discuss the proposed methodology in detail and validate its accuracy via experiments and comparisons with other MF methods. Section 5 presents the conclusions and scope of future work; finally, the reviewed materials are referenced, where many of which are recent publications.

2. Related Work

Several studies in the recent past have harnessed hierarchical and tag information as auxiliary features to address issues related to data sparsity and cold starts in recommendation systems [13,25,28,29]. CF-based recommender systems are commonly employed to predict ratings based on user histories; however, they ignore costly features, which introduce data sparsity and cold starts, which in turn hampers performance. Therefore, various studies have integrated auxiliary information in the recommendation process [30,31].

Auxiliary features often maintain a rich knowledge structure i.e., a hierarchy with dependencies. Yang et al. [13] proposed an MF-based framework with recursive regularization that analyzes the impacts of hierarchically organized features in user–item interactions to improve the recommendation accuracy and eliminate the cold-start problem. Lu et al. [32] developed a framework that exploited these hierarchical relationships to identify more reliable neighbors; moreover, the framework modeled the hierarchical structure based on potential users' preferences. The hierarchical itemspace rank (HIR) algorithm utilizes the intrinsic hierarchical structure of an itemspace to mitigate data sparsity that may affect the quality of recommendations [33].

Most modern recommender systems trawl both explicit and implicit data for useful information, including ratings, images, text (tags), social information, items, and user characteristics, to offer recommendations. We can thus infer that analyzing tag information is important in recommender systems, as they not only recap the characteristics of items but also help in identifying user preferences. For example, food recommendations are made by a model trained on a dataset comprising user preferences that are collated from ratings and tags specified in product forms to indicate their preferred food components and features [25]. Karen et al. [27] proposed a generic method that modifies CF algorithms to accommodate tags and deconstructs 3D correlations into three 2D correlations. Moreover,

Wang et al. [34] formulated a novel approach that combined tags and ratings-based CF to discern similar users and items.

Our proposed methodology deviates from these methods in that the tags obtained from user–item interactions are used to regularize the MF process, whereas hierarchical information delivers the rating predictions. In summary, existing MF models that use hierarchical and tag information individually have delivered satisfactory results despite the complexity. However, to the best of our knowledge, there is no available advantageous work that seamlessly incorporates the hierarchical and tag information.

3. Methodology

This section is devoted to illustrating our proposed methodology that predicts rating scores by evolving hierarchical structures of items and users simultaneously with a mathematically modeled combination of tag information. Specifically, the notations that are used in this paper are first introduced, and then, a basic model that builds the basis of the proposed model is described. After that, we go into the details of the model components that mathematically model the hierarchical structures of items and users simultaneously and the integration of tag information, respectively, the combination of which leads to an optimization problem. Lastly, we come up with an efficient algorithm to solve it.

3.1. Notations

Table 1 enumerates the notations used in this paper.

Table 1. Notation definitions.

Notation	Description
H	Matrices are denoted by boldface capital letters
h	Vectors are denoted by boldface lowercase letters
$\ H\ _F$	Frobenius norm of matrix
\odot	Hadamard product
λ	Regularization parameter
$\text{tr}(\cdot)$	Trace of a matrix
β	Extra regularization parameter

3.2. Basic Matrix Factorization

We modeled our approach on a basic weighted non-negative matrix factorization (WNMF) method owing to its feasible and easy implementation in recommendation systems with large inputs and sparse data. This method factorizes an input rating matrix into two non-negative sub-matrices **P** and **Q** of sizes $n \times r$ and $r \times m$, respectively.

$$\mathbf{R}' \approx \mathbf{P}\mathbf{Q} = \begin{bmatrix} p_1 \\ p_2 \\ \dots \\ p_n \end{bmatrix} \begin{bmatrix} q_1 & q_2 & \dots & q_m \end{bmatrix} \quad (1)$$

The rating score given by p_i to q_j is then computed as $\mathbf{R}'(i,j) = \mathbf{P}(i,:) \mathbf{Q}(:,j)$. **P** and **Q** are evaluated by solving the following optimization problem:

$$\min_{\mathbf{P}, \mathbf{Q}} \|W \odot (\mathbf{R} - \mathbf{P}\mathbf{Q})\|_F^2 + \lambda (\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2) \quad (2)$$

where W is the hyperparameter that regulates the contribution of $\mathbf{R}'(i, j)$ in the learning process such that $W(i, j) = 1$ for $\mathbf{R}'(i, j) > 0$; else, $W(i, j) = 0$. \odot is the Hadamard element-wise multiplication operator, λ is the regularization parameter used to moderate the complexity and overfitting during learning, and $\|\mathbf{P}\|_F^2$ and $\|\mathbf{Q}\|_F^2$ are the Frobenius norms of the corresponding matrices [27].

3.3. Acquiring the Hierarchical Structured Information

Some features of users and items are hierarchically structured. For instance, as shown in Figure 1b, the genres of movies can be organized into a hierarchical structure. It is very likely that movies that are associated with the detailed genres are more similar than those in subgenres. For this reason, it should be suitable to recommend a movie that is in the same detailed genre as one that has got a high rating score from the user. Hierarchical structures of users and items involve complementary information and capturing them simultaneously can further improve the recommendation performance. Therefore, in this subsection, acquiring the hierarchically structured information of users and items is introduced by enhancing the basic weighted non-negative matrix factorization model.

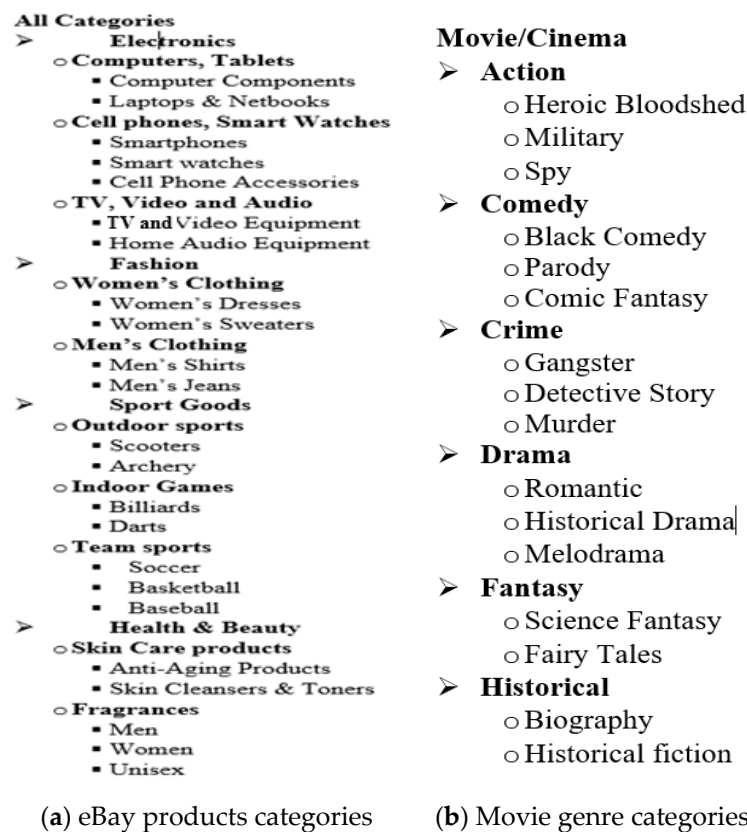


Figure 1. (a) Hierarchical structure of eBay products and (b) an illustration of movie genre categories.

One of the most significant challenges for recommendation systems is to elicit valuable information from the features of highly correlated users and items in a user–item interaction that forms the basis of the prediction process. Typically, this is modeled using the flat attributes of users (for example, gender and age) or items (in the case of a movie, this can include an actor, a producer, release date, language, and country). However, these features may often be represented in a multilevel structure, i.e., a hierarchy, in the form of a tree with nested nodes (for example, movie genres and user occupations). Simple representations of a hierarchical structure include movie genres and product categories on e-commerce websites, as shown in Figure 1.

For example, the movie *Godfather* (an item) can be classified by traversing the hierarchical tree nodes as follows: main genre→subgenre, per Figure 1b, which then resembles crime→gangster.

Similarly, the Apple Watch Series 5 (an item) can be placed in a hierarchical structure, per Figure 1a, as main category → subcategory → explicit subcategory, which is tantamount to electronics → cell phones, smart watches → smart watches. The classification of an item into appropriate lower-level categories or nodes is conducted sequentially.

User preferences are similarly structured. For instance, a user who chooses to rate movies in the crime genre may prefer the gangster subgenre over others, and those who shop for items belonging to a particular hierarchical level of the product catalog may express coincidental preferences by consistently rating items that exhibit similar characteristics.

From Section 3.2, WNMF was adopted as the core model to acquire implicit hierarchical information and thereby predict rating scores. The user–item rating matrix, \mathbf{R} , was deconstructed into two lower-dimensional non-negative submatrices, \mathbf{P} and \mathbf{Q} , constituting user preferences and item characteristics, respectively, and expressed as the flat structures of features. Because \mathbf{P} and \mathbf{Q} are non-negative, we applied the non-negative matrix factorization to them to interpret the corresponding hierarchically structured information, which then served to predict the rating scores given by Equation (1).

\mathbf{P} and \mathbf{Q} were extracted such that $\mathbf{P} \in \mathbb{R}^{n \times r}$ and $\mathbf{Q} \in \mathbb{R}^{r \times m}$ to indicate the latent representations of n users and m items in an r -dimensional latent category (space). \mathbf{P} and \mathbf{Q} were further factorized to model the hierarchical structure owing to their non-negativity.

Therefore, in a particular embodiment, \mathbf{P} was factorized into two matrices, $\mathbf{P}_1 \in \mathbb{R}^{n \times n_1}$ and $\tilde{\mathbf{P}}_2 \in \mathbb{R}^{n_1 \times r}$, as follows:

$$\mathbf{P} \approx \mathbf{P}_1 \tilde{\mathbf{P}}_2 \quad (3)$$

where n is the number of users, r is the number of latent categories (space) in the first hierarchical level, and n_1 is the number of subcategories in the second hierarchical level. Thus, $\mathbf{P}_1 \in \mathbb{R}^{n \times n_1}$ is the relationship of n users to n_1 subcategories. $\tilde{\mathbf{P}}_2$ denotes the second level of the hierarchical structure of users obtained from the relationship between the number of latent categories (space) in the first hierarchical level and n_1 , i.e., the number of latent subcategories in the second hierarchical level. To compute the third level of a hierarchical structure of users, as given in Equation (4), $\tilde{\mathbf{P}}_2$ is further factorized as $\mathbf{P}_2 \in \mathbb{R}^{n_1 \times n_2}$ and $\tilde{\mathbf{P}}_3 \in \mathbb{R}^{n_2 \times r}$:

$$\mathbf{P} \approx \mathbf{P}_1 \mathbf{P}_2 \tilde{\mathbf{P}}_3 \quad (4)$$

where n_2 is the number of subcategories in the third hierarchical level. Therefore, deep factorization on \mathbf{P} serves to obtain the x th level of the hierarchical structure of users, \mathbf{P}_x , which is accomplished by factorizing $\tilde{\mathbf{P}}_{x-1}$, the latent category relationship matrix of the $(x-1)$ th level of the hierarchical structure, into non-negative matrices, as follows:

$$\mathbf{P} \approx \mathbf{P}_1 \mathbf{P}_2 \dots \mathbf{P}_{x-1} \mathbf{P}_x \quad (5)$$

where $\mathbf{P}_i \geq 0$ for $i \in \{1, 2, \dots, x\}$, \mathbf{P}_1 is an $n \times n_1$ matrix such that \mathbf{P}_i is an $n_{i-1} \times n_i$ matrix, and \mathbf{P}_x is $n_{x-1} \times r$ matrix.

The above factorization process as illustrated in Figure 2 is repeated for \mathbf{Q} to obtain the level of the hierarchical structure of items. For this, the relationship of m items with r -dimensional latent categories (space) is represented as $\mathbf{Q} \in \mathbb{R}^{r \times m}$, which is further factorized into $\mathbf{Q}_1 \in \mathbb{R}^{m_1 \times m}$ and $\tilde{\mathbf{Q}}_2 \in \mathbb{R}^{r \times m_1}$ to describe the second level of items in the hierarchy given by:

$$\mathbf{Q} \approx \tilde{\mathbf{Q}}_2 \mathbf{Q}_1 \quad (6)$$

where m_1 is the number of sub-categories in the second hierarchical level and $\mathbf{Q}_1 \in \mathbb{R}^{m_1 \times m}$ is the relationship of m items to the m_1 latent subcategories. The latent category relationship of the

non-negative matrix $\tilde{\mathbf{Q}}_2 \in \mathbb{R}^{r \times m_1}$ of the second hierarchical level is defined as the affiliation between r -dimensional latent categories (space) in the first hierarchical level and m_1 latent subcategories in the second hierarchical level. Equation (7) gives the third level of the hierarchical structure of items, where $\tilde{\mathbf{Q}}_2$ is also factorized as $\mathbf{Q}_2 \in \mathbb{R}^{m_2 \times m_1}$ and $\tilde{\mathbf{Q}}_3 \in \mathbb{R}^{r \times m_2}$, where m_2 is the number of subcategories in the third hierarchical level:

$$\mathbf{Q} \approx \tilde{\mathbf{Q}}_3 \mathbf{Q}_2 \mathbf{Q}_1 \quad (7)$$

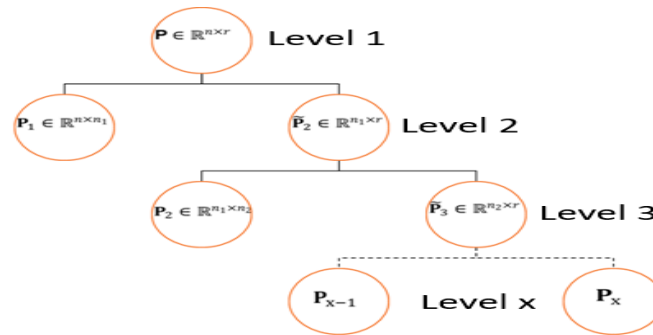


Figure 2. Obtaining the hierarchical structure of users.

Deep factorization on \mathbf{Q} , as illustrated in Figure 3, secures the y th level of a hierarchical structure of items, \mathbf{Q}_y , which is accomplished by factorizing $\tilde{\mathbf{Q}}_{y-1}$, in the $(y-1)$ th level of the hierarchy, as follows:

$$\mathbf{Q} \approx \mathbf{Q}_y \mathbf{Q}_{y-1} \dots \mathbf{Q}_2 \mathbf{Q}_1 \quad (8)$$

where $\mathbf{Q}_j \geq 0$ for $j \in \{1, 2, \dots, y\}$, \mathbf{Q}_1 is an $m_1 \times m$ matrix such that \mathbf{Q}_j is an $m_j \times m_{j-1}$ matrix, and \mathbf{Q}_y is an $r \times m_{y-1}$ matrix.

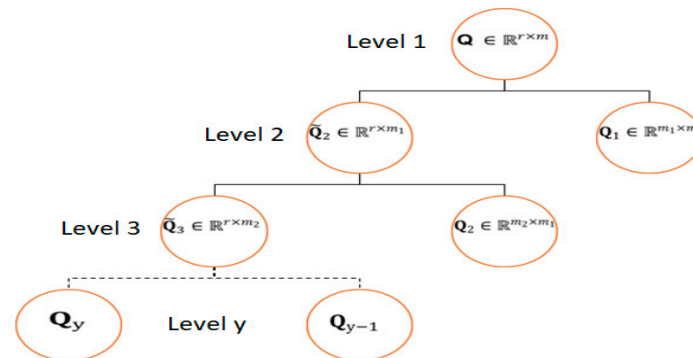


Figure 3. Obtaining a hierarchical structure of items.

Finally, the below optimization problem needs to be effectively solved for building a model that outlines the hierarchical structures of users and items:

$$\min_{\mathbf{P}_1, \dots, \mathbf{P}_x, \mathbf{Q}_1, \dots, \mathbf{Q}_y} \|\mathbf{W} \odot (\mathbf{R} - \mathbf{P}_1 \dots \mathbf{P}_x \mathbf{Q}_y \dots \mathbf{Q}_1)\|_F^2 + \lambda \left(\sum_{i=1}^x \|\mathbf{P}_i\|_F^2 + \sum_{j=1}^y \|\mathbf{Q}_j\|_F^2 \right) \quad (9)$$

where $\mathbf{P}_i \geq 0$ for $i \in \{1, 2, \dots, x\}$ and $\mathbf{Q}_j \geq 0$ for $j \in \{1, 2, \dots, y\}$.

The rating prediction process that involves acquired user's and item's hierarchically structured information is represented in Figure 4.

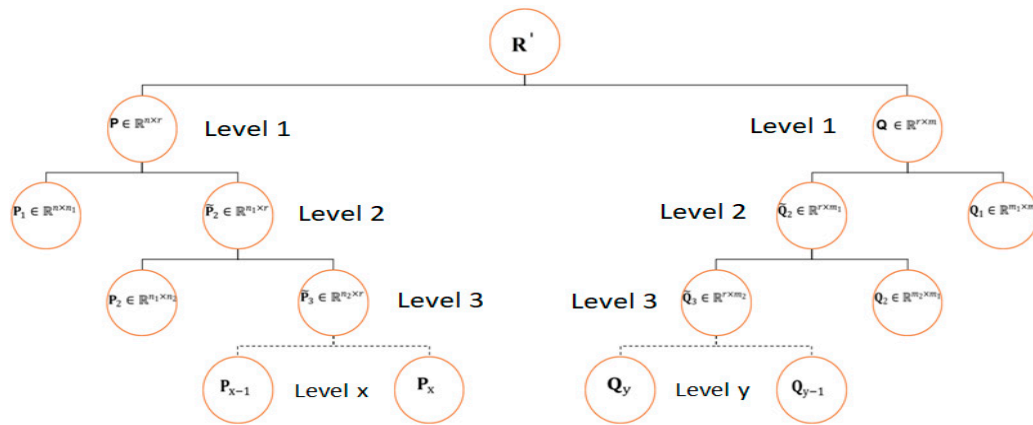


Figure 4. An illustration of predicting a rating score based on hierarchical structures of users and items.

3.4. Incorporating Tag Information

Tag information was incorporated uniquely into our proposed methodology for deriving an association between the supplementary information solicited from WNMF and tag repetitiveness in items [3]. For example, an “organized crime” tag assigned to the movie “The Godfather” (item) by a user may also apply to other items with similar characteristics, which is reflected in the degree of repetitiveness. Therefore, the matrix factorization process of a basic WNMF model is regularized using the tag information to complete our prediction model. In short, we aimed to form two item-specific latent feature vectors from the MF process of our WMNF model that are similar in nature and contain items with common tag information. For a tag information matrix T , each of its components T_{it} for item i and tag t is a $tf * idf$ value [35]:

$$T_{it} = tf(i, t) * \log_2 \left(\frac{m}{df(t)} \right) \quad (10)$$

where $tf(i, t)$ is the normalized frequency of t occurring in i , $df(t)$ is the number of items that contain t , and m is the total number of items. Thus, the similarity between items i and j is computed using the cosine similarity metric given, as follows:

$$S_{i,j} = \frac{\sum_{t \in T^{ij}} T_{it} T_{jt}}{\sqrt{\sum_{t \in T^{ij}} T_{it}^2} \sqrt{\sum_{t \in T^{ij}} T_{jt}^2}} \quad (11)$$

where T^{ij} is the index of tags occurring in both i and j . The two item-specific latent feature vectors that are most similar are then obtained by affixing an item similarity regularization criterion function to the WNMF model, as follows:

$$\begin{aligned} \frac{\beta}{2} \sum_{i=1}^N \sum_{j=1}^N S_{i,j} \|q_i - q_j\|_F^2 &= \frac{\beta}{2} \sum_{i=1}^N \sum_{j=1}^N \left[S_{i,j} \sum_{r'=1}^r (q_{r'i} - q_{r'j})^2 \right] \\ &= \frac{\beta}{2} \sum_{r'=1}^r \mathbf{Q}_{r'^*} \mathbf{L} \mathbf{Q}_{r'^*}^T = \frac{\beta}{2} \text{tr}(\mathbf{Q} \mathbf{L} \mathbf{Q}^T) \end{aligned} \quad (12)$$

where $S_{i,j}$ defines the similarity between i and j ; q_1, q_2, \dots, q_m are latent characteristic vectors that populate \mathbf{Q} ; r is the dimension of each item in the vector, i.e., $q_{r'i}$ and $q_{r'j}$ are the values of vector items i and j of the r' th dimension; \mathbf{L} denotes the Laplacian matrix given by $\mathbf{L} = \mathbf{D} - \mathbf{S}$ for a diagonal matrix \mathbf{D} such that $D_{ij} = \sum_j S_{ij}$. $\text{tr}(\cdot)$ is a trace of the matrix; β is an extra regularization parameter that controls the contribution of the tag information [36].

The rating predictions were made by combining Equations (9) and (12) and utilizing the following objective function for the minimization task:

$$\min_{\mathbf{P}_1, \dots, \mathbf{P}_x, \mathbf{Q}_1, \dots, \mathbf{Q}_y} \|W \odot (\mathbf{R} - \mathbf{P}_1 \dots \mathbf{P}_x \mathbf{Q}_y \dots \mathbf{Q}_1)\|_F^2 + \lambda \left(\sum_{i=1}^x \|\mathbf{P}_i\|_F^2 + \sum_{j=1}^y \|\mathbf{Q}_j\|_F^2 \right) + \frac{\beta}{2} \text{tr}(\mathbf{Q} \mathbf{L} \mathbf{Q}^T) \quad (13)$$

where $\mathbf{P}_i \geq 0$ for $i \in \{1, 2, \dots, x\}$ and $\mathbf{Q}_j \geq 0$ for $j \in \{1, 2, \dots, y\}$.

3.5. Optimization Problem

The optimization problem is complicated owing to the non-convexity of the objective function, but solving for it also helps in validating the method that is administered in a recommendation system. Our optimization method modified the approach in [37] in that all variables of the objective function given in Equation (13) were updated interchangeably such that the function becomes convex, which does not occur otherwise.

3.5.1. The Basis of Updating \mathbf{P}_i

When \mathbf{P}_i is updated, terms unrelated to \mathbf{P}_i are discarded by fixing the other variables, and the resulting objective function is expressed as:

$$\min_{\mathbf{P}_i \geq 0} \|W \odot (\mathbf{R} - \mathbf{A}_i \mathbf{P}_i \mathbf{H}_i)\|_F^2 + \lambda \|\mathbf{P}_i\|_F^2 \quad (14)$$

where \mathbf{A}_i and \mathbf{H}_i for $1 \leq i \leq x$, are defined as:

$$\mathbf{A}_i = \begin{cases} \mathbf{P}_1 \mathbf{P}_2 \dots \mathbf{P}_{x-1} & \text{if } i \neq 1 \\ \mathbf{I} & \text{if } i = 1 \end{cases} \quad (15)$$

$$\mathbf{H}_i = \begin{cases} \mathbf{P}_{i+1} \dots \mathbf{P}_x \mathbf{Q}_y \dots \mathbf{Q}_1 & \text{if } i \neq x \\ \mathbf{Q}_y \dots \mathbf{Q}_1 & \text{if } i = x \end{cases} \quad (16)$$

The Lagrangian function in Equation (14) is:

$$L(\mathbf{P}_i) = \|W \odot (\mathbf{R} - \mathbf{A}_i \mathbf{P}_i \mathbf{H}_i)\|_F^2 + \lambda \|\mathbf{P}_i\|_F^2 - \text{Tr}(\mathbf{M}^T \mathbf{P}_i) \quad (17)$$

where \mathbf{M} is the Lagrangian multiplier. The derivative of $L(\mathbf{P}_i)$ with respect to \mathbf{P}_i is then given by:

$$\frac{\partial L(\mathbf{P}_i)}{\partial \mathbf{P}_i} = 2\mathbf{A}_i^T \|W \odot (\mathbf{A}_i \mathbf{P}_i \mathbf{H}_i - \mathbf{R})\|_F^T + 2\lambda \mathbf{P}_i - \mathbf{M} \quad (18)$$

By setting the derivative to zero and employing the Karush–Kuhn–Tucker complementary condition [37], i.e., $\mathbf{M}(s, t) \mathbf{P}_i(s, t) = 0$, we obtain:

$$[\mathbf{A}_i^T [W \odot (\mathbf{A}_i \mathbf{P}_i \mathbf{Q} - \mathbf{R})] \mathbf{H}_i^T + \lambda \mathbf{P}_i](s, t) \mathbf{P}_i(s, t) = 0 \quad (19)$$

Finally, the updated rule of \mathbf{P}_i is computed using:

$$\mathbf{P}_i(s, t) \leftarrow \mathbf{P}_i(s, t) \sqrt{\frac{[\mathbf{A}_i^T (W \odot \mathbf{R}) \mathbf{H}_i^T](s, t)}{[\mathbf{A}_i^T (W \odot (\mathbf{A}_i \mathbf{P}_i \mathbf{H}_i)) \mathbf{H}_i^T + \lambda \mathbf{P}_i](s, t)}} \quad (20)$$

3.5.2. The Basis of Updating Q_i

Similarly, for Q_i , the unrelated terms are initially discarded by fixing the other variables, and the resulting objective function is expressed as:

$$\min_{Q_i \geq 0} \|W \odot (R - B_i Q_i K_i)\|_F^2 + \lambda \|Q_i\|_F^2 + \frac{\beta}{2} \text{tr}(QLQ^T) \quad (21)$$

where B_i and K_i for $1 \leq i \leq x$, are defined as:

$$B_i = \begin{cases} P_1 \dots P_x Q_y \dots Q_{y+1} & \text{if } i \neq y \\ P_1 \dots P_x & \text{if } i = y \end{cases} \quad (22)$$

$$K_i = \begin{cases} Q_{y-1} \dots Q_1 & \text{if } i \neq 1 \\ I & \text{if } i = 1 \end{cases} \quad (23)$$

We can then compute the updated rule for Q_i in the same way as P_1 :

$$Q_i(s, t) \leftarrow Q_i(s, t) \sqrt{\frac{[B_i^T (W \odot R) K_i^T + \frac{\beta}{2} \text{tr}(QLQ^T)](s, t)}{[B_i^T (W \odot (B_i Q_i K_i)) K_i^T + \lambda Q_i + \frac{\beta}{2} \text{tr}(QLQ^T)](s, t)}} \quad (24)$$

The optimization with the above updating rules for P_i and Q_j tries to unveil the approximation of the factors in the proposed model. Each hierarchical level is pre-trained to get an initial approximation of the matrices P_i and Q_j . The input user-item rating matrix is factorized into $\tilde{P}_1 \tilde{Q}_1$ by solving Equation (2). Then, \tilde{P}_1 and \tilde{Q}_1 are further factorized into $\tilde{P}_1 \approx P_1 \tilde{P}_2$ and $\tilde{Q}_1 \approx \tilde{Q}_2 Q_1$, respectively. The factorization step is continued up until the p th user and q th item hierarchical levels are obtained. The fine-tuning process is performed by updating P_i and Q_i using Equations (20) and (24) separately. The step first involves updating Q_i in sequence and then P_i in sequence. Finally, the predicted rating matrix will be equal to $R' = P_1 \dots P_x Q_y \dots Q_1$.

3.6. Convergence Analysis

The examination of the convergence of the proposed model was conducted as follows.

The assistant function in [38] was used to prove the convergence of the model.

Definition 1. The assistant function [38] is defined as $G(h, h')$ for $F(h)$ if the conditions:

$$G(h, h') \geq F(h), \quad G(h, h) = F(h) \quad (25)$$

are satisfied.

Assumption 1. If G [38] is an assistant function for F , then F is non-increasing under the update:

$$h^{(t+1)} = \arg \min G(h, h^{(t)}) \quad (26)$$

Proof.

$$F(h^{(t+1)}) \leq G(h^{(t+1)}, h^{(t)}) \leq G(h^{(t)}, h^{(t)}) \leq F(h^{(t)}) \quad (27)$$

□

Assumption 2. [39] For any matrices $A \in \mathbb{R}_+^{n \times n}$, $B \in \mathbb{R}_+^{k \times k}$, $S \in \mathbb{R}_+^{k \times k}$, and $S' \in \mathbb{R}_+^{k \times k}$, where A and B are symmetric, the following inequality holds:

$$\sum_{s=1}^n \sum_{t=1}^k \frac{(AS'B)(s,t)S^2(s,t)}{S'(s,t)} \geq \text{Tr}(S^T ASB) \quad (28)$$

The objective function in Equation (14) can be written in the following form by developing the quadratic terms and removing terms that are unrelated to P_i :

$$J(P_i) = \text{Tr}(-2A_i^T(W \odot R)H_i^T P_i^T) + \text{Tr}(A_i^T(W \odot (A_i^T P_i H_i))H_i^T P_i^T) + \text{Tr}(\lambda P_i P_i^T) \quad (29)$$

Theorem 1.

$$\begin{aligned} G(P, P') &= -2 \sum_{s,t} (A_i^T(W \odot R)H_i^T)(s,t) P_i(s,t) \left(1 + \log \frac{P_i(s,t)}{P'_i(s,t)}\right) \\ &+ \sum_{s,t} \frac{(A_i^T(W \odot (A_i^T P_i H_i))H_i^T)(s,t) P_i^2(s,t)}{P'_i(s,t)} + \text{Tr}(\lambda P_i P_i^T) \end{aligned} \quad (30)$$

The above function is an assistant function for $J(P_i)$. Moreover, it is a convex function in (P_i) and its global minimum is:

$$P_i(s,t) \leftarrow P_i(s,t) \sqrt{\frac{[A_i^T(W \odot R)H_i^T](s,t)}{[A_i^T(W \odot (A_i P_i H_i))H_i^T + \lambda P_i](s,t)}} \quad (31)$$

Proof. The proof is similar to that in [40] and thus the details are omitted. \square

Theorem 2. Updating P_i with Equation (20) will monotonically decrease the value of the objective in Equation (13).

Proof. With Assumption 1 and Theorem 1, we have:

$$J(P_i^{(0)}) = G(P_i^{(0)}, P_i^{(0)}) \geq G(P_i^{(1)}, P_i^{(0)}) \geq J(P_i^{(1)}) \quad (32)$$

That is, $J(P_i)$ decreases monotonically. Equivalently, the update rule for Q_i will also monotonically decrease the value of the objective in Equation (13). Since the value of the objective in Equation (13) is at least edged by zero, we can have shown that the optimization technique of the proposed method converges. \square

3.7. Time Complexity Analysis

The most expensive operations in the proposed model are the initialization and fine-tuning process that leads to increasing the efficiency of the model. Namely, the time complexity of the decomposition of $\tilde{P}_i \in \mathbb{R}^{n_{i-1} \times r}$ to $P_i \in \mathbb{R}^{n_{i-1} \times n_i}$ and $\tilde{P}_{i+1} \in \mathbb{R}^{n_i \times r}$ is $O(kn_{i-1}n_i r)$ for $1 < i < x$ and $O(kn_1 r)$ for $i = 1$, where k is the number of iterations in the decomposition process. Hence, the cost of initializing the P_i 's is $O(kr(nn_1 + n_1 n_2 + \dots + n_{x-2} n_{x-1}))$. Likewise, the cost of initializing the Q_i 's is $O(kr(mm_1 + m_1 m_2 + \dots + m_{y-2} m_{y-1}))$. The computational costs of fine-tuning P_i and Q_i in each iteration are $O(nn_{i-1}n_i + nn_i m + n_{i-1}n_i m)$ and $O(mm_{i-1}m_i + mm_i n + m_{i-1}m_i n)$. Let $n_0 = n$, $m_0 = m$, $n_x = m_y = r$, then the time complexity of fine-tuning is $O(k_f \left[(n+m) \left(\sum_{i=1}^x n_{i-1}n_i + \sum_{j=1}^y m_{j-1}m_j \right) + nm \left(\sum_{i=1}^x n_i + \sum_{j=1}^y m_j \right) \right])$, where k_f is the number of iterations in the fine-tuning process. The time complexity of computing the item similarities and L is $O(m^2 t)$, where m is the total number of items and t is the total number of tags. Hence, the total time complexity is the sum of the cost of the initialization, fine-tuning, and computing the item similarities. It is interesting to note that in practice, two hierarchical levels of users and items, $x = 2$ and $y = 2$, give better performance advancement over MF and WNMF. When $x > 2$ and $y > 2$, the performance of

the proposed model is also better than that of $x = 2$ and $y = 2$, but the time complexity grows. Therefore, the optimal value of x and y is chosen to be 2 practically because the time complexity is not larger than for MF and WNMF.

4. Experiment

4.1. Dataset

To evaluate the performance of our model, an experiment was performed with the latest small MovieLens 100K dataset. The dataset comprises 100,000 movie ratings and 3683 tags that are essentially user-generated metadata (a single word or short phrase) about movies. The ratings are scored on a scale of 0.5 to 5.0 stars, and movies and users are selected from a total of 19 genres and 21 occupation categories, respectively. While the genres and occupations are leveraged for hierarchical information of the movies and users, the tags lend to tag information.

4.2. Measurement Metric

The dataset was randomly divided into 60% and 80% for training, and the remaining instances were split as 40% and 20% for testing. The prediction accuracy of the proposed model was measured using the popular mean absolute error (MAE) metric. MAE returns the average absolute deviation of the prediction from the ground truth:

$$\text{MAE} = \frac{\sum_{(i,j) \in \tau} |R_{ij} - R'_{ij}|}{|\tau|} \quad (33)$$

where τ is a set of ratings, and R and R' are the true and predicted ratings, respectively. The smaller the value of MAE, the more accurate the prediction; hence, MAE is preferred when the indicator values are small.

4.3. Results

We evaluated the model using two indicators: the rating prediction error (i.e., MAE) for the predetermined weights of the tag information and the extent of mitigating the item cold-start problem.

4.3.1. Prediction Accuracy with Tag Information Weights

It is worth noting that the proposed method completed the entire workflow for the rating prediction only in the case of items constituting tag information, while for the rest of the instances, it morphed into a basic WNMF model, i.e., without solving for Equations (10)–(13). To prove the superiority of our approach, two baseline methods were selected for comparison, where the results are summarized in Table 2.

1. Matrix factorization: Proposed by Koren et al. [3], this method factorizes a user–item rating matrix and learns the resultant user and item latent feature vectors to minimize the error between the true and predicted ratings.
2. Weighted non-negative matrix factorization: This was also chosen as the base model for the proposed approach, where WNMF attempts to factorize a weighted user–item rating matrix into two non-negative submatrices to minimize the error between the true and predicted ratings.

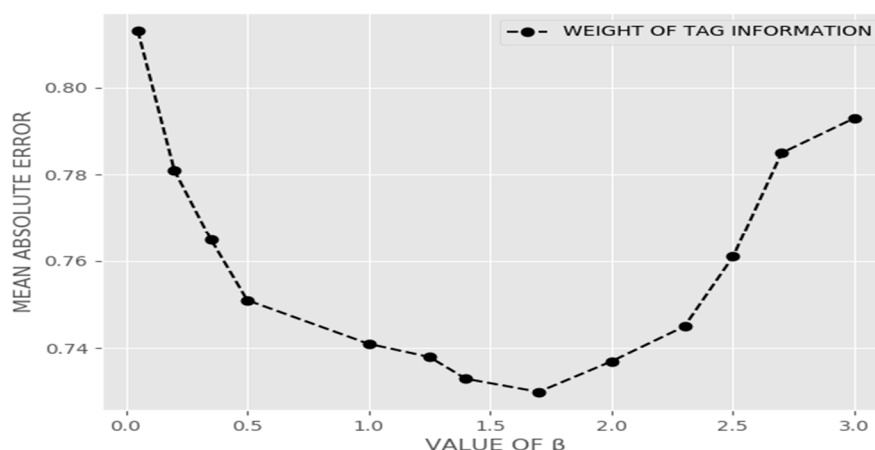
The results were taken when the parameter r was defined as 20 and the size of n_1 ranged according to {50, 100, 150, 200, 250}, while m_1 ranged according to {100, 200, 300, 400, 500}. The values of the hierarchical layers x and y were equal to 2. $W \odot R \approx W \odot (P_1 P_2 Q_1 Q_2)$ where the matrices are given as follows $P_1 \in \mathbb{R}^{n \times n_1}$, $P_2 \in \mathbb{R}^{n_1 \times r}$ and $Q_1 \in \mathbb{R}^{r \times m_1}$, $P_2 \in \mathbb{R}^{m_1 \times m}$. Overall, when the values of the dimensions rose, the model performance tended to grow at first and consequently fell.

Table 2. Comparison of the mean absolute error (MAE) results of the rating predictions between different methods and the proposed approach.

Training Set Size (%)	MAE				
	MF	WNMF	MAE	Proposed	
				Number of Optimal Hierarchical Levels	
				x (Users)	y (Items)
60	0.7635	0.7820	0.7386	2	2
80	0.7586	0.7657	0.7309	2	2

MF: matrix factorization, WNMF: weighted non-negative matrix factorization.

The extra regularization parameter, β , controls the contribution of tag information in learning the item latent feature vector. In other words, for $\beta = 0$, our methodology adopted the basic WNMF to compute Equation (13) and thereafter predict ratings, whereas, for non-zero β values, the weight of the tag information manifested its effects on the predictions, as illustrated in Figure 5. Although this reflects a certain degree of reliance on β for the proposed approach, it also proved the efficiency of using a combination of hierarchical and tag information. The correlation between MAE and β for β in the range of 0.05–3.0 was plotted for the 80% training dataset and the accuracy increased proportionally with β , peaking between 1.0 and 2.1 (lowest MAE recorded).

**Figure 5.** The weight of the tag information in the recommendation system.

4.3.2. Mitigation of the Item Cold Start

One of the main challenges encountered when building a recommendation system is the cold-start problem, which arises when a new user or item is introduced for which no past interactions are available. In particular, collaborative filtering algorithms are more prone to the cold-start problem. As basic models, matrix factorization algorithms (WNMF and MF) have poor performances in the case of the cold-start problem due to a lack of preference information [26,27,41]. Supposing the tag information is accessible for use, our proposed model can mitigate the cold-start problem by seamlessly incorporating the tag information to provide a recommendation. Tag information not only contains an explanation of the items but also provides the sentiment of users. In particular, the proposed method tries to make two item-specific latent feature vectors as similar as possible if the two items have a similar tagging history. It can give recommendations to new users who have no preference for any items. In such cases, the proposed approach helped in alleviating the cold-start problem by integrating tag information, where other comparable methods failed.

To test this, the ratings of 50 and 100 randomly selected items from the 80% training dataset were discarded such that they were viewed as new items (cold-start items) in the recommendation system. In the cold-start experiments, the results of the proposed model performance were taken when the parameters of the model were set to the optimal values of $\beta = 1.8$, $r = 20$, and the number of hierarchical layers x and y were equal to 2. The comparative results are presented in Table 3, which shows that the proposed method outperformed the MF and WNMf models, validating the conducted test, showing that tag information could be used to execute recommendations for cold-start items. It is evident that in both instances, the proposed methodology helped with mitigating the cold-start problem for new items significantly better than its competitors.

Table 3. MAE performance comparisons for the item cold-start problem.

Cold-Start Case	50 Cold-Start Items					100 Cold-Start Items				
	MF	WNMF	MAE	Proposed		MF	WNMF	MAE	Proposed	
				Number of Optimal Hierarchical Levels					Number of Optimal Hierarchical Levels	
				x (Users)	y (Items)				x (Users)	y (Items)
All items	0.8894	0.8461	0.8096	2	2	0.9135	0.8836	0.8740	2	2
Cold-start items	0.9247	0.8613	0.8287	2	2	0.9591	0.9165	0.9107	2	2

4.3.3. Top-N Recommendation Results

Along with providing superior MAE results for rating predictions, the proposed model also showed its superiority when performing the top-N recommendation task. Experiments on the proposed model for top-N recommendation identified the items that best fit the user's personal tastes obtained from their hierarchically structured features and tagging history. To evaluate the top-N performance of the proposed model, an 80% training dataset was used to generate a ranked list of size N items for each user. The proposed method and the other two baseline cutting edge methods were compared using the most widely used MovieLens 100K dataset, as indicated in Figure 6. The comparison task was performed for three sizes of N : the first was the top-5, the second was the top-10, and the final one was the top-15. When the size of N was equal to 5, the MAE of the MF method was 0.748, while the MAE of the WNMf method was higher by 0.01 than the MF method. However, the proposed model outperformed both the MF and WNMf methods and accomplished the lowest error rate of 0.736 for the top-5 and 0.752 for the top-10, whereas the other two methods (MF and WNMf) showed 0.757 and 0.772 for the top-10, respectively. Our suggested approach required expensive operations for the initialization and fine-tuning process. For this reason, the proposed method had a slightly higher error rate compared to the MF method, as indicated for the top-15. From these experiments, the proposed method still worked successfully and the superiority was clearly verified.

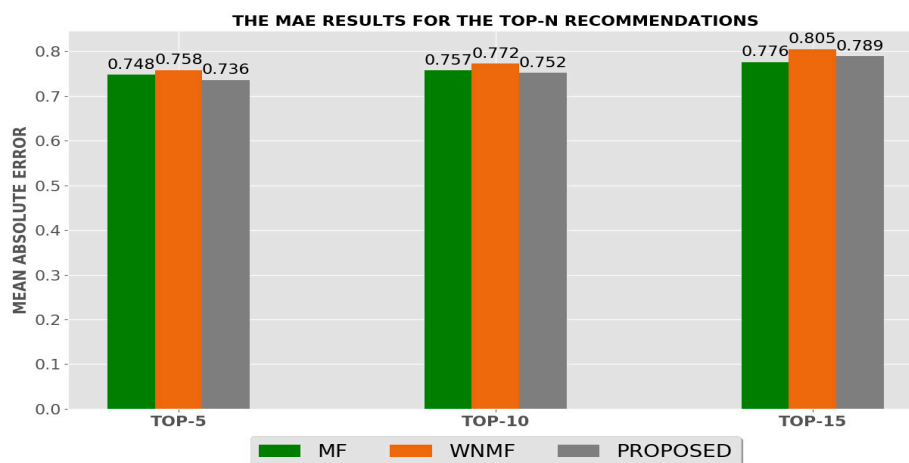


Figure 6. The MAE results for the top-N recommendations.

5. Conclusions

Presently, while the development of personalized recommendation systems has been continuing to grow to a high degree, data sparsity, cold starts, and improving recommendation system performances are still open challenges that need to be solved in the recommendation system area. In this study, we proposed a novel rating prediction model with enhanced matrix factorization using hierarchical and tag information that addressed the above issues. Experimental results revealed the significant influence of the hierarchical and tag information used in combination to alleviate the issues of data sparsity and item cold starts compared to established MF techniques. The entire workflow of our proposed model for rating predictions was completed only in the case of items constituting tag information with the hierarchical information of users and items. In particular, deep factorization on the user preference and item characteristic matrices was accomplished due to their non-negativity to get hidden-level hierarchical structured features, while tag information was used to regularize the matrix factorization process of a basic WNMf model to complete our prediction model. During the experimental testimony process, we concluded that if the values of the dimensions increased, the proposed model performance tended to increase at first and then decrease. Despite the superiority of the proposed approach, several problems were encountered, especially with the advances in the domain that focus on the high volume of data available for making recommendations. Therefore, future research could explore more sophisticated models for estimating the importance of the hidden features of users and items that the features represented as hierarchical structures, as well as tag information preference, by using recent deep learning methods and algorithms. Additionally, future research work might similarly also develop an explainable and interpretable recommendation system based on the above hidden features.

Author Contributions: This manuscript was designed and written and the experiments were performed by A.K. A.A. helped to revise and improve the manuscript. The theory and experiments were analyzed and commented on by T.K.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by Ministry of Culture, Sports and Tourism and Korea Creative Content Agency (Project Number: R2020040243).

Acknowledgments: The authors A.K. and A.A. would like to express their sincere gratitude and appreciation to the supervisor, Taeg Keun Whangbo (Gachon University) for his support, comments, remarks, and engagement over the period in which this manuscript was written. Moreover, the authors would like to thank the editor and anonymous referees for the constructive comments in improving the contents and presentation of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bobadilla, J.; Ortega, F.; Hernando, A.; Gutierrez, A. Recommender systems survey. *Knowl. Based Syst.* **2013**, *46*, 109–132. [\[CrossRef\]](#)
2. Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P.B. *Recommender Systems Handbook*; Springer: Berlin, Germany, 2011; ISBN 978-0-387-85819-7.
3. Koren, Y.; Bell, R.; Volinskiy, C. Matrix factorization techniques for recommender systems. *IEEE Comput.* **2009**, *42*, 30–37. [\[CrossRef\]](#)
4. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep Learning based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* **2018**, *52*, 1–38. [\[CrossRef\]](#)
5. Ortega, F.; Hurtado, R.; Bobadilla, J.; Bojorquez, R. Recommendation to groups of users the singularities concept. *IEEE Access* **2018**, *6*, 39745–39761. [\[CrossRef\]](#)
6. Tuzhilin, A.; Adomavicius, G. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 734–749.
7. Su, X.; Khoshgoftaar, T.M. A survey of collaborative filtering techniques. *Adv. Artif. Intell.* **2009**. [\[CrossRef\]](#)
8. Chatti, M.A.; Dakova, S.; Thus, H.; Schroeder, U. Tag-based collaborative filtering recommendation in personal learning environments. *IEEE Trans. Learn. Technol.* **2012**, *6*, 337–349. [\[CrossRef\]](#)
9. Goldberg, D.; Nichols, D.; Oki, B.M.; Terry, D. Using collaborative filtering to weave an information tapestry. *Commun. ACM* **1992**, *35*, 61–70. [\[CrossRef\]](#)

10. Liu, J.; Tang, M.; Zheng, Z.; Liu, X.; Lyu, S. Location-Aware and Personalized Collaborative Filtering for Web Service Recommendation. *IEEE Trans. Serv. Comput.* **2016**, *9*, 686–699. [[CrossRef](#)]
11. Herlocker, J.L.; Konstan, J.A.; Borchers, A.; Riedl, J. An algorithmic framework for performing collaborative filtering. In Proceedings of the SIGIR'99: 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, USA, 15–19 August 1999; pp. 230–237.
12. Guo, X.; Yin, S.-C.; Zhang, Y.-W.; Li, W.; He, Q. Cold start recommendation based on attribute-fused singular value decomposition. *IEEE Access* **2019**, *7*, 11349–11359. [[CrossRef](#)]
13. Yang, J.; Sun, Z.; Bozzon, A.; Zhang, J. Learning hierarchical feature influence for recommendation by recursive regularization. In Proceedings of the Recsys: 10th ACM Conference on Recommender System, Boston, MA, USA, 15–19 September 2016; pp. 51–58.
14. Koren, Y.; Bell, R. Advances in collaborative filtering. In *Recommender Systems Handbook*; Springer: Berlin, Germany, 2011; pp. 145–186.
15. *Unifying User-Based and Item-Based Collaborative Filtering Approaches by Similarity Fusion*; SIGIR '06; ACM: New York, NY, USA, 2006.
16. Zarei, M.R.; Moosavi, M.R. A Memory-Based Collaborative Filtering Recommender System Using Social Ties. In Proceedings of the 4th International Conference on Pattern Recognition and Image Analysis (IPRIA), Tehran, Iran, 6–7 March 2019.
17. Stephen, S.C.; Xie, H.; Rai, S. Measures of similarity in memory-based collaborative filtering recommender system: A comparison. In Proceedings of the 4th Multidisciplinary International Social Networks Conference, 4th Multidisciplinary International Social Networks Conference (MISNC), Bangkok, Thailand, 17–19 July 2017.
18. Al-bashiri, H.; Abdulgaber, M.A.; Romli, A.; Kahtan, H. An improved memory-based collaborative filtering method based on the TOPSIS technique. *PLoS ONE* **2018**, *13*, e0204434. [[CrossRef](#)] [[PubMed](#)]
19. Li, X.; Li, D. An Improved Collaborative Filtering Recommendation Algorithm and Recommendation Strategy. *Mobile Inform. Syst.* **2019**. [[CrossRef](#)]
20. Fang, Y.; Si, L. Matrix co-factorization for recommendation with rich side information and implicit feedback. In *Hetrec 11*; ACM: New York, NY, USA, 2011.
21. Kumar, A.; Soder, N. Open problems in recommender systems diversity. In Proceedings of the International Conference on Computing, Communication and Automation (ICCCA2017), Greater Noida, India, 5–6 May 2017.
22. Salakhutdinov, R.; Mnih, A. Probabilistic matrix factorization. In Proceedings of the NIPS'07: 20th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2007.
23. Seo, S.; Huang, J.; Yang, H.; Liu, Y. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Recsys '17*; ACM: New York, NY, USA, 2017.
24. Maleszka, M.; Mianowska, B.; Nguyen, N.T. A method for collaborative recommendation using knowledge integration tools and hierarchical structure of user profiles. *Knowl. Based Syst.* **2013**, *47*, 2013. [[CrossRef](#)]
25. Ge, M.; Elahi, M.; Tobias, I.F.; Ricci, F.; Massimo, D. Using tags and latent factors in a food recommender system. In Proceedings of the DH '15: 5th International Conference on Digital Health, Florence, Italy, 18–20 May 2015.
26. Garg, N.; Weber, I. Personalized, interactive tag recommendation for flickr. In Proceedings of the 2nd ACM International Conference on Recommender Systems, RecSys'08, Lausanne, Switzerland, 23–25 October 2008; pp. 67–74. [[CrossRef](#)]
27. Tso-Sutter, K.H.L.; Marinho, L.B.; Schmidt-Thieme, L. Tag-aware recommender systems by fusion collaborative filtering algorithms. In Proceedings of the SAC '08: 2008 ACM Symposium on Applied Computing, Fortaleza, Brazil, 16–20 March 2008.
28. Schein, A.I.; Popescul, A.; Ungar, L.H.; Pennock, D.M. Methods and metrics for cold-start recommendations. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, 11–15 August 2002; pp. 253–260.
29. Vall, A.; Skowron, M.; Schedl, M. *Improving Music Recommendations with a Weighted Factorization of the Tagging Activity*; ISMIR: Montreal, QC, Canada, 2015.
30. Shi, C.; Liu, J.; Zhuang, F.; Yu, P.S.; Wu, B. Integrating Heterogeneous Information via Flexible Regularization Framework for Recommendation. *Knowl. Inform. Syst.* **2016**, *49*, 835–859. [[CrossRef](#)]

31. Wu, J.; Chen, L.; Yu, Q.; Han, P.; Wu, Z. *Trust-Aware Media Recommendation in Heterogeneous Social Networks*; Springer: Berlin, Germany, 2015.
32. Lu, K.; Zhang, G.; Li, R.; Zhang, S.; Wang, B. Exploiting and exploring hierarchical structure in music recommendation. In *AIRS 2012: Information Retrieval Technology*; Springer: Berlin, Germany, 2012; pp. 211–225.
33. Nikolakopoulos, N.; Kouneli, M.A.; Garofalakis, J.D. Hierarchical Itemspace Rank: Exploiting hierarchy to alleviate sparsity in ranking-based recommendation. *J. Neurocomput.* **2015**, *163*, 126–136. [[CrossRef](#)]
34. Wang, Z.; Wang, Y.; Wu, H. Tag meet ratings: Improving collaborative filtering with tag-based neighborhood method. In Proceedings of the SRS'10 ACM, Hong Kong, China, 7 February 2010.
35. Shepitsen, A.; Gemmell, J.; Mobasher, M.; Burke, R. Personalized recommendation in social tagging systems using hierarchical clustering. In Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys, Lausanne, Switzerland, 23–25 October 2008.
36. Chung, F. *Spectral Graph Theory*; American Mathematical Society: Providence, RI, USA, 1997.
37. Trigeorgis, G.; Bousmalis, K.; Zaferiou, S.; Schuller, B. A deep semi-nmf model for learning hidden representations. In Proceedings of the 31st International Conference on Machine Learning (ICML-14), Beijing, China, 21–26 June 2014; pp. 1692–1700.
38. Lee, D.D.; Seung, H.S. Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.* **2001**, *13*, 556–562.
39. Ding, C.; Li, T.; Peng, W.; Park, H. Orthogonal nonnegative matrix t-factorizations for clustering. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006; pp. 126–135.
40. Gu, Q.; Zhou, J.; Ding, C.H.Q. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In Proceedings of the 2010 SIAM International Conference on Data Mining, Columbus, OH, USA, 29 April–1 May 2010; pp. 199–210.
41. Lam, X.N.; Vu, T.; Le, T.D.; Duong, A.D. Addressing cold-start problem in recommendation systems. In Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, Suwon, Korea, 31 January 2008; pp. 208–211.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).